Gated Differential Linear Attention: A Linear-Time Decoder for High-Fidelity Medical Segmentation

Supplementary Material

6. Preliminaries

6.1. Multi-Head Softmax Attention

Let the input be $\boldsymbol{X} \in \mathbb{R}^{N \times d_{\text{model}}}$, where N is the number of tokens (e.g., $N = \mathcal{H}\mathcal{W}$ for a 2D grid) and d_{model} is the channel width. The softmax attention [36] operates with the following steps.

Input Linear Projections. The input $\boldsymbol{X}$ is linearly transformed into queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$, and values $\boldsymbol{V} \in \mathbb{R}^{N \times d_k}$ via learned weight matrices $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$,

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}^Q, \quad \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}^K, \quad \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}^V. \tag{29}$$

Scaled Dot-Product (Softmax) Attention. The scaled dot-product attention (SDPA) is computed as

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\intercal}{\sqrt{d_k}}\right)\boldsymbol{V} \in \mathbb{R}^{N \times d_k}, \tag{30}$$

where $\boldsymbol{Q}\boldsymbol{K}^\intercal/\sqrt{d_k}$ is the similarity matrix and $\text{softmax}(\cdot)$ normalizes each row into a probability distribution.

Multi-Head Concatenation. In multi-head attention, this computation is performed in parallel across h heads. Each head uses separate projection matrices $\boldsymbol{W}_i^Q, \boldsymbol{W}_i^K, \boldsymbol{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_h}$ for $i \in \{1, \ldots, h\}$, yielding

$$\text{head}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V), \tag{31}$$

with $\text{head}_i \in \mathbb{R}^{N \times d_h}$. The multi-head output is then

$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h). \tag{32}$$

Output Linear Projection. The concatenated multi-head output is linearly projected with a learnable matrix $\boldsymbol{W}^O \in \mathbb{R}^{hd_h \times d_{\text{model}}}$,

$$\boldsymbol{O} = \text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\boldsymbol{W}^O. \tag{33}$$

7. Experiment Details

This appendix complements Section 4 of the main paper with additional details on datasets, implementation settings, evaluation metrics, and supplementary experimental results.

7.1. Dataset and Implementation Details

Synapse Multi-Organ. We follow the TransUNet protocol [8] on *Synapse* (30 CT scans), using 18 scans for training and 12 for validation. Training runs for 200 epochs with 10 warmup epochs, batch size 8, and AdamW [21] with learning rate 5×10^{-4}.

Automated Cardiac Diagnosis Challenge (ACDC). On *ACDC* [4] (100 cardiac MRI scans), we split 70/10/20 for train/val/test. Training uses 150 epochs with 10 warmup epochs, batch size 16, and AdamW with learning rate 5×10^{-4}.

Breast Ultrasound Images Dataset (BUSI). For *BUSI* [1], we adopt an 80/10/10 train/val/test split. Training uses 100 epochs with 10 warmup epochs, batch size 16, and AdamW with learning rate 5×10^{-4}.

HAM10000. For *HAM10000* [35] (10,015 dermoscopic images), we use a 70/10/20 train/val/test split. Training uses 150 epochs with 10 warmup epochs, batch size 16, and AdamW with learning rate 5×10^{-4}.

PH^2. On *PH*^{2} [24] (200 images), the split is 80 train, 20 validation, and 100 test images. Training uses 100 epochs with 5 warmup epochs, batch size 16, and AdamW with learning rate 5×10^{-4}.

7.2. Evaluation Metrics

We use Dice score (DSC) to evaluate performance on all the datasets. For the *Synapse* multi-organ benchmark, we additionally report the 95th-percentile Hausdorff distance (HD95), which is commonly used to evaluate boundary quality in medical image segmentation.

Dice Score. The Dice score is computed as

$$\text{DSC}(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \tag{34}$$

where X is the prediction and Y is the ground truth.

Hausdorff Distance. The Hausdorff Distance is computed as

$$d_H(X, Y) = \max\{\max_{x \in X} \min_{y \in Y} d(x, y), \max_{y \in Y} \min_{x \in X} d(x, y)\} \tag{35}$$

where X is the prediction and Y is the ground truth.

7.3. Results

In this section, we provide additional qualitative results from different datasets and analysis of the attention in the proposed gated differential linear attention (GDLA) mixer.

7.3.1. Attention Visualization

Figure 8 compares GDLA mixer with differential attention (DA), self-attention (SA), and linear attention (LA) across decoder stages. The change-in-attention maps ($\|\Delta\text{attn}\|$) show consistent qualitative differences across the four variants. LA produces broadly elevated and spatially diffuse updates, which is consistent with attention dilution. DA sharpens responses relative to LA, but still exhibits spurious high-frequency activations in background regions. SA generates more localized updates, yet occasionally concentrates excessively on a small number of tokens and incurs higher computational cost. In contrast, GDLA concentrates update energy around organ interiors and boundaries, suppresses background activations, and maintains a more balanced response across decoder stages. These observations are qualitatively consistent with the Dice improvements reported in the main paper.

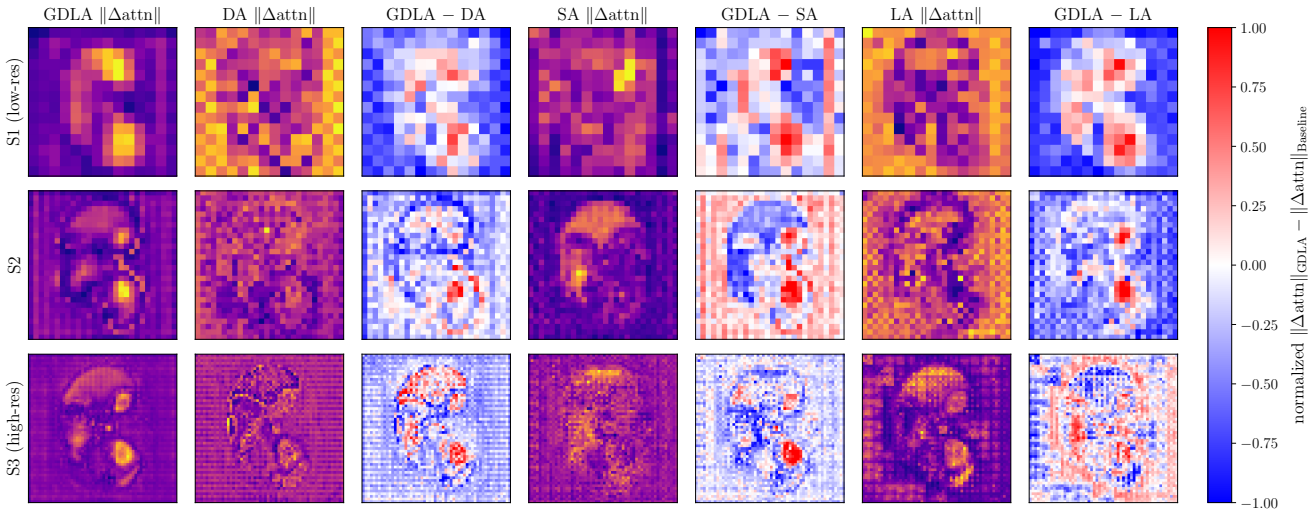

Figure 8. GDLA vs. baseline attentions on $\|\Delta\text{attn}\|$. Rows show decoder stages (S1–S3, coarse → fine). Columns display per-method update magnitudes for GDLA, Differential Attention (DA), Self Attention (SA), and Linear Attention (LA), alongside normalized difference maps ($\|\Delta\text{attn}\|_{\text{GDLA}} - \|\Delta\text{attn}\|_{\text{baseline}}$).